200

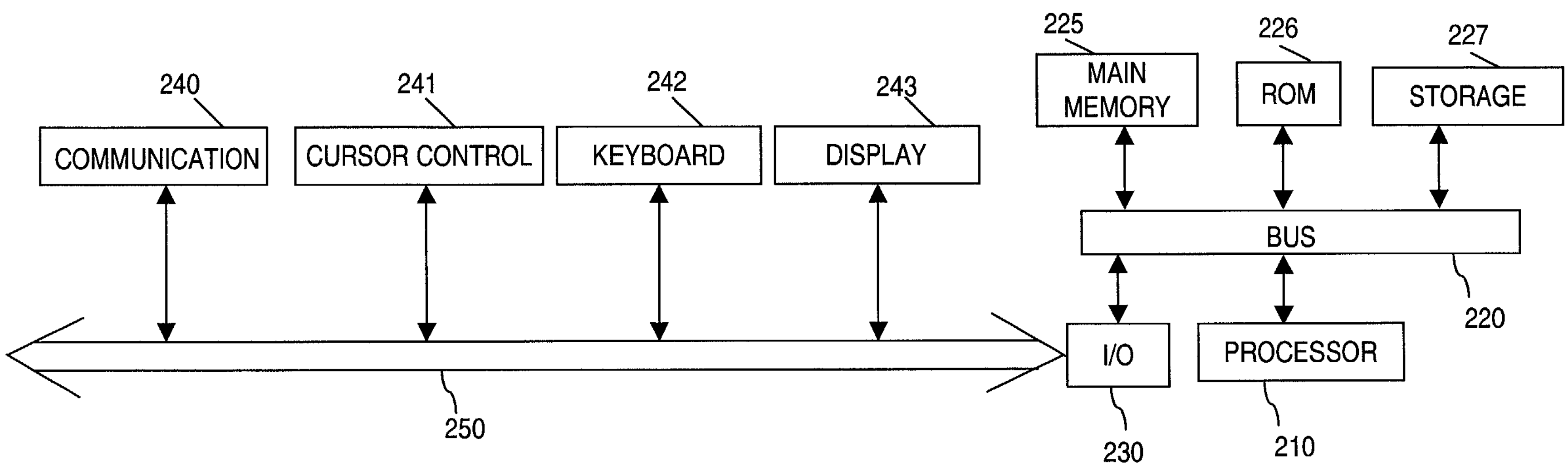

240
COMMUNICATION
241
CURSOR CONTROL
242
KEYBOARD
243
DISPLAY
225
MAIN MEMORY
226
ROM
227
STORAGE
BUS
220
I/O
PROCESSOR
250
230
210

FIG. 1

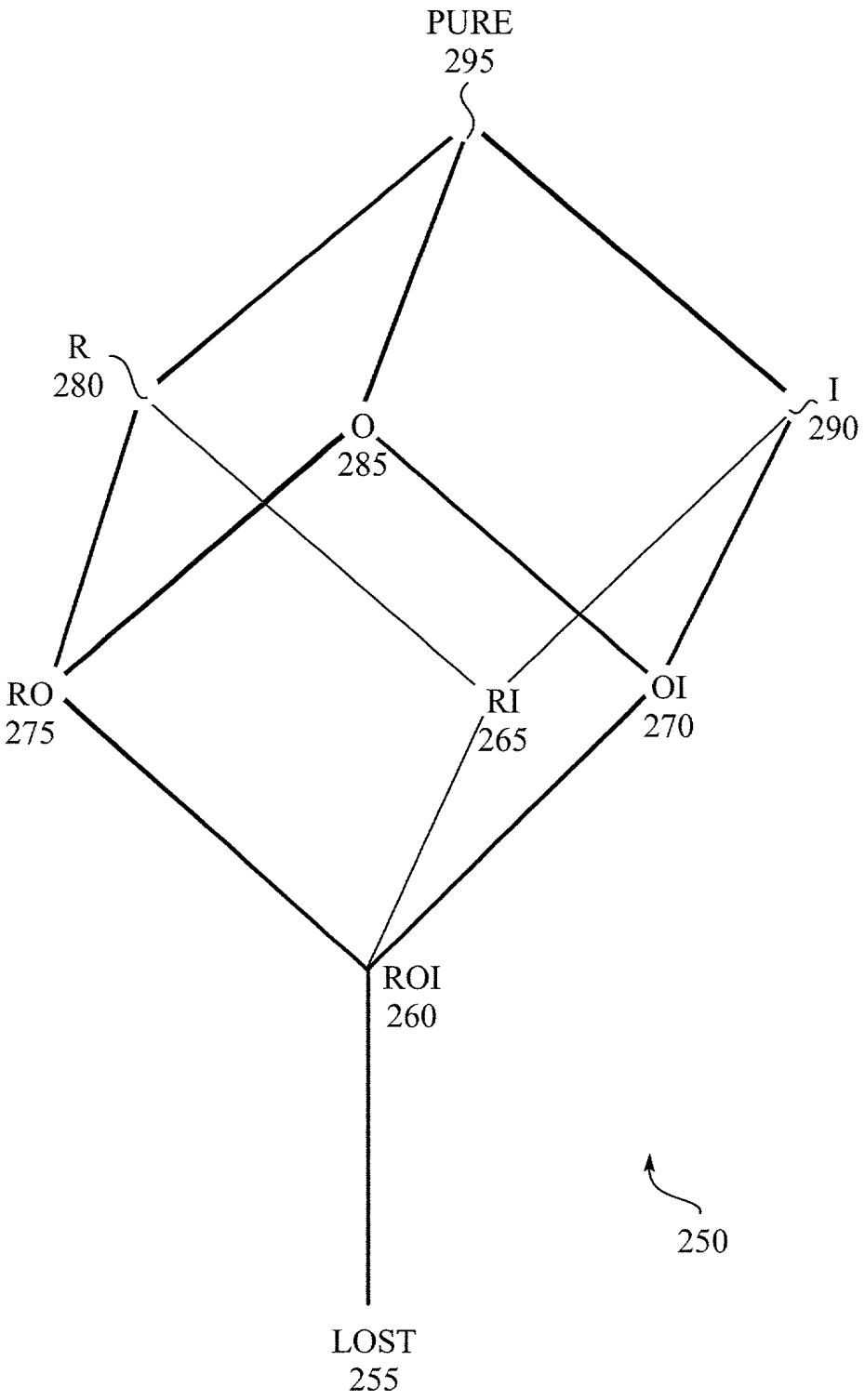

PURE
295
R
280
O
285
I
290
RO
275
RI
265
OI
270
ROI
260
LOST
255
250

Fig. 2

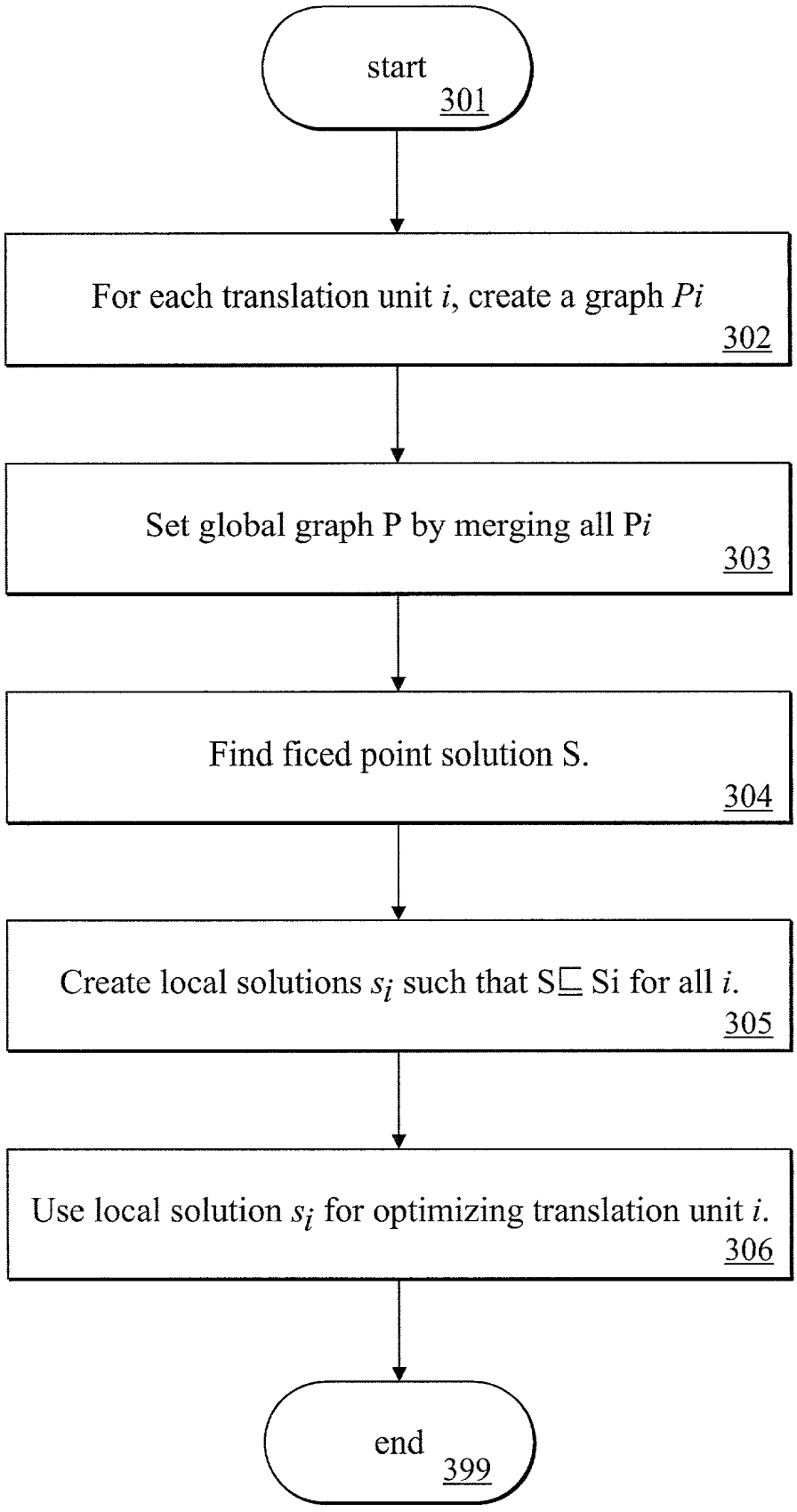

start
301
For each translation unit i, create a graph Pi
302
Set global graph P by merging all Pi
303
Find ficed point solution S.
304
Create local solutions s_i such that S⊑ Si for all i.
305
Use local solution s_i for optimizing translation unit i.
306
end
399

Fig. 3

Fujnction	Fujnction (x,y)
410 ~ TOP	(PURE,PURE)
420 ~ COPY	*(y,y)*
430 ~ IN_TO_LOST	if $y \leq I \Rightarrow$ (LOST,LOST) otherwise ⇒ (PURE,PURE)
440 ~ UNRETURN	if y=LOST⇒(LOST,LOST) otherwise ⇒ $(z,z)$where $z=y \sqcup OI$
450 ~ COPY_AND_IN_TO_LOST	if $y \leq I \Rightarrow$ (LOST,LOST) otherwise ⇒ (y,y) (y,PURE)
460 ~ CAT_FORMAL	(PURE,y)
470 ~ CAT_ACTUAL	if x=LOST⇒ (LOST,LOST) else if x<R (z,z) where $z = (x \sqcup OI) \sqcap y$ else$(z,z)$where $z = (x \sqcup OI)$
480 ~ GATE	

Fig. 4A

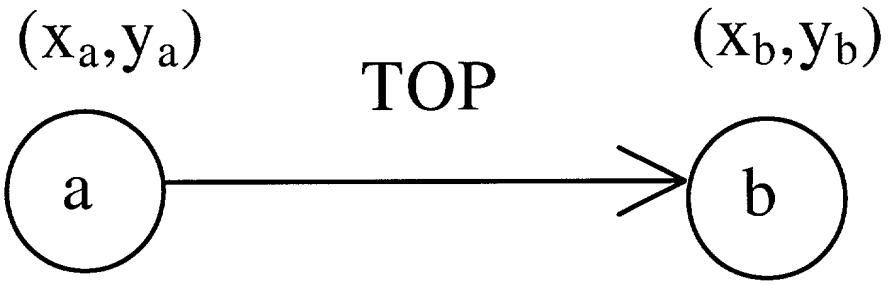


FIG. 4B

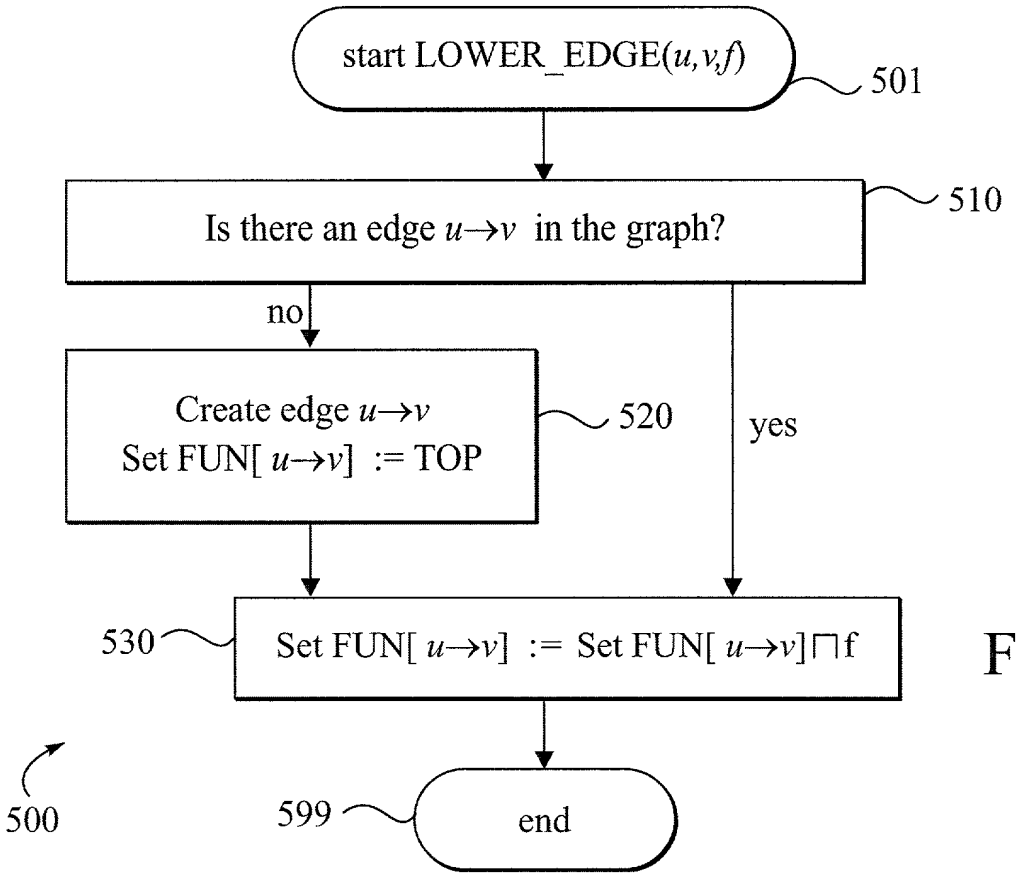

start LOWER_EDGE(u,v,f)
501
Is there an edge u→v in the graph?
510
no
yes
Create edge u→v
Set FUN[u→v] := TOP
520
530
Set FUN[u→v] := Set FUN[u→v]⊓f
500
599
end

Fig. 5

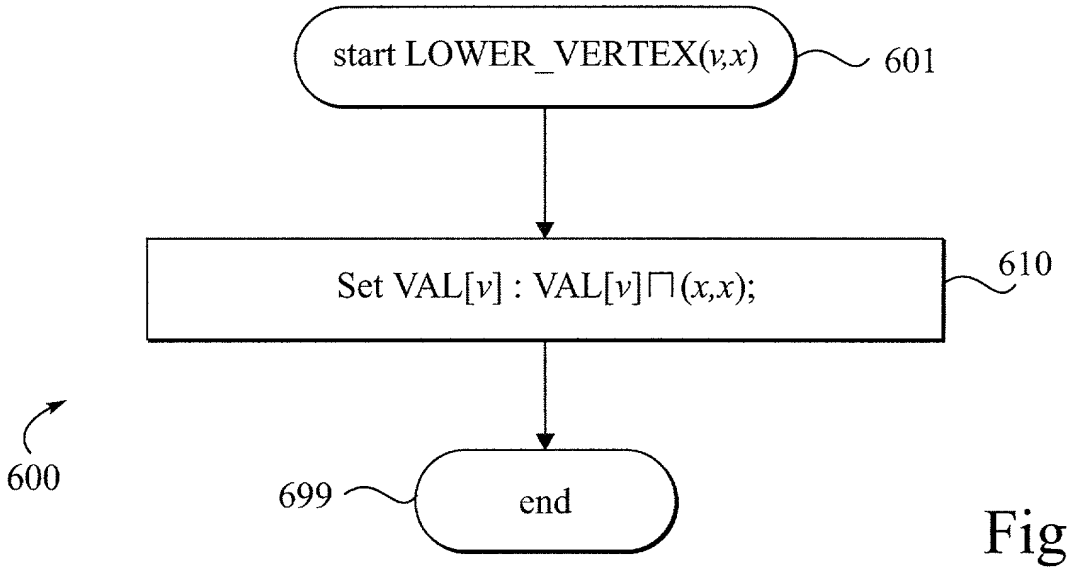

start LOWER_VERTEX(v,x)
601
Set VAL[v] : VAL[v]⊓(x,x);
610
600
699
end

Fig. 6

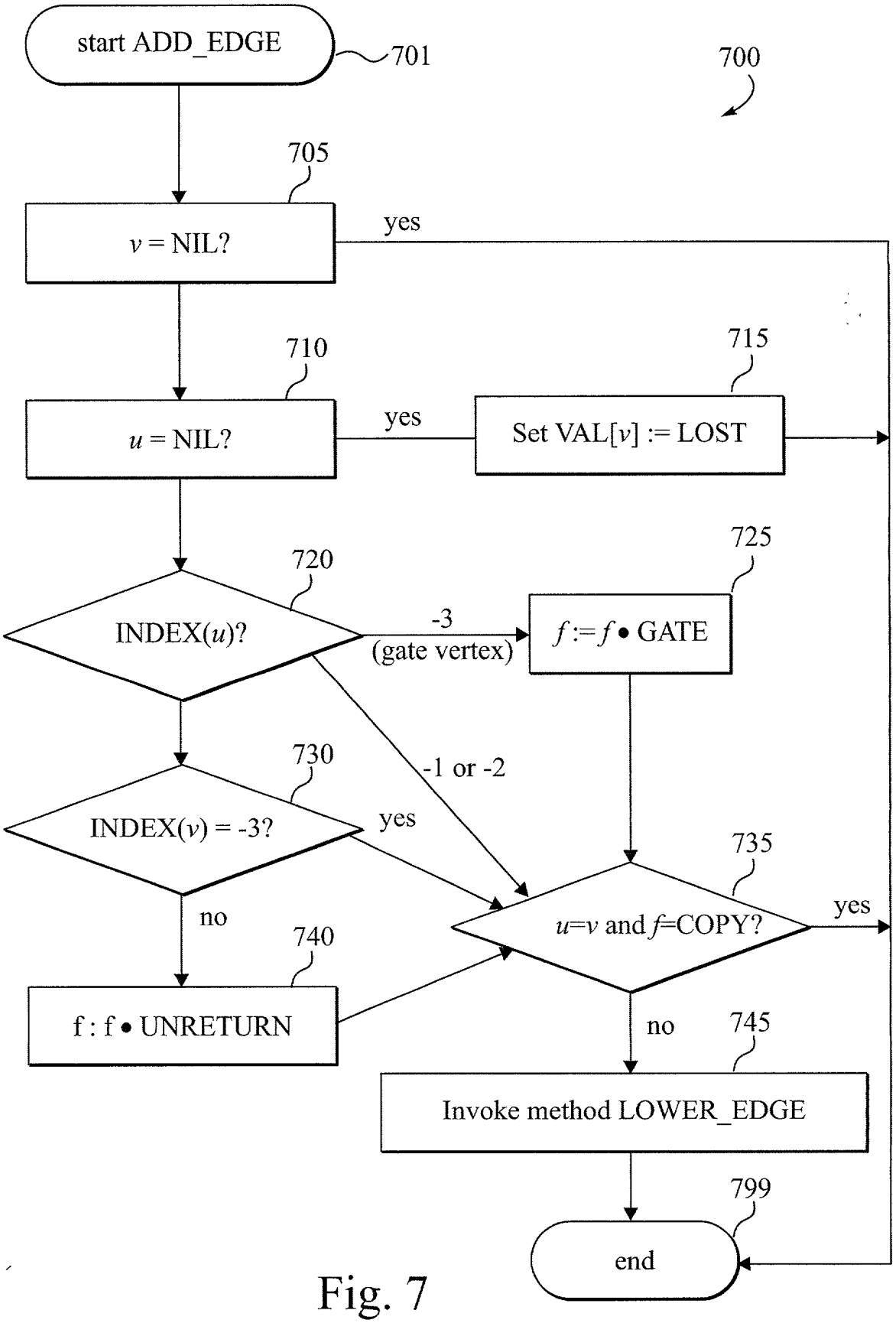

start ADD_EDGE
701
700
705
v = NIL?
yes
710
715
u = NIL?
yes
Set VAL[v] := LOST
720
725
INDEX(u)?
-3
(gate vertex)
f := f • GATE
730
-1 or -2
INDEX(v) = -3?
yes
735
u=v and f=COPY?
yes
no
740
f : f • UNRETURN
no
745
Invoke method LOWER_EDGE
799
end

Fig. 7

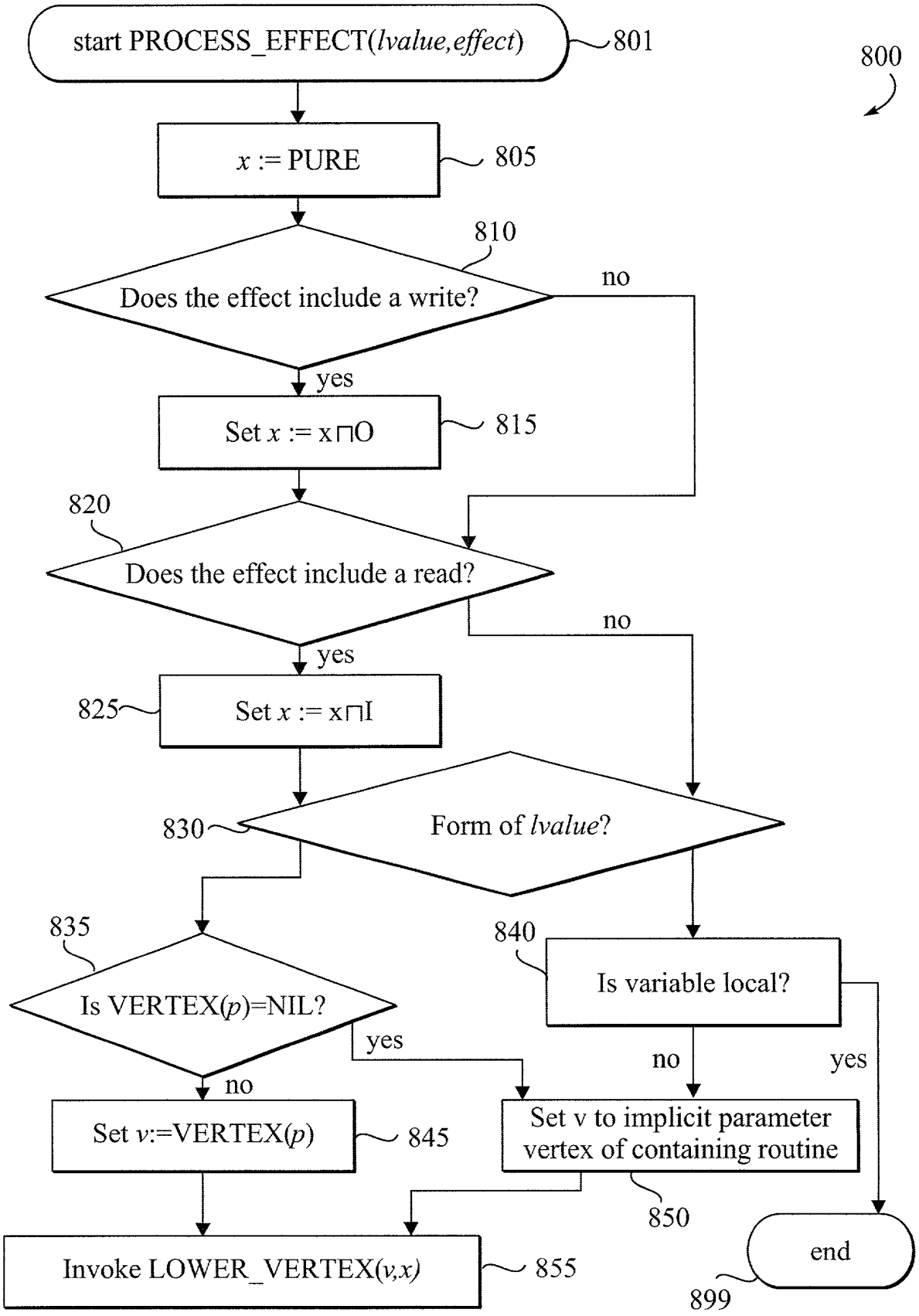

start PROCESS_EFFECT(*lvalue,effect*)
801
800
x := PURE
805
810
Does the effect include a write?
no
yes
Set *x* := x⊓O
815
820
Does the effect include a read?
no
yes
825
Set *x* := x⊓I
830
Form of *lvalue*?
835
Is VERTEX(*p*)=NIL?
840
Is variable local?
yes
no
yes
no
Set *v*:=VERTEX(*p*)
845
Set v to implicit parameter vertex of containing routine
850
Invoke LOWER_VERTEX(*v,x*)
855
end
899

Fig. 8

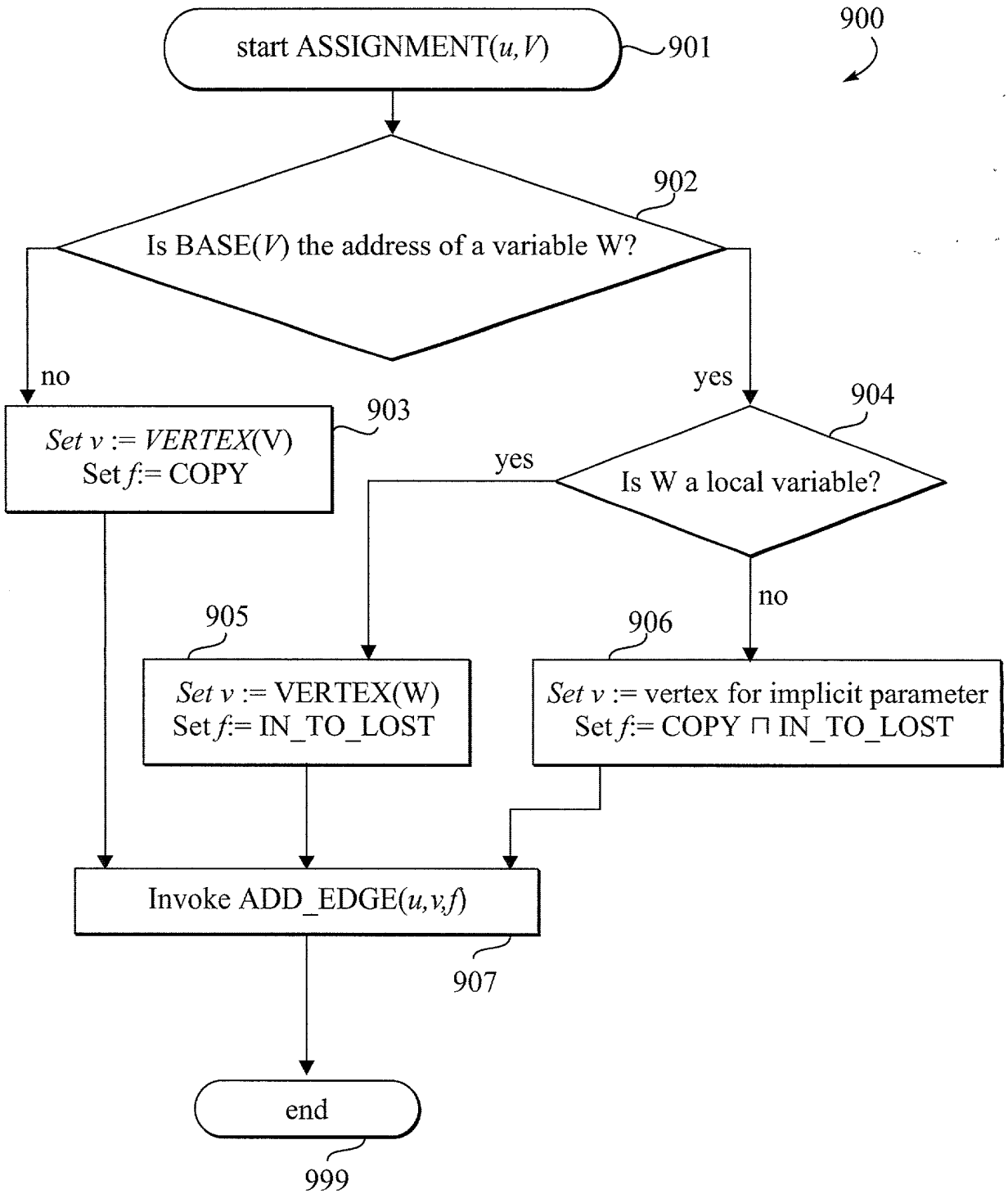

900
start ASSIGNMENT(u, V)
901
902
Is BASE(V) the address of a variable W?
no
yes
903
Set v := VERTEX(V)
Set f:= COPY
904
yes
Is W a local variable?
no
905
Set v := VERTEX(W)
Set f:= IN_TO_LOST
906
Set v := vertex for implicit parameter
Set f:= COPY ⊓ IN_TO_LOST
Invoke ADD_EDGE(u,v,f)
907
end
999

Fig. 9

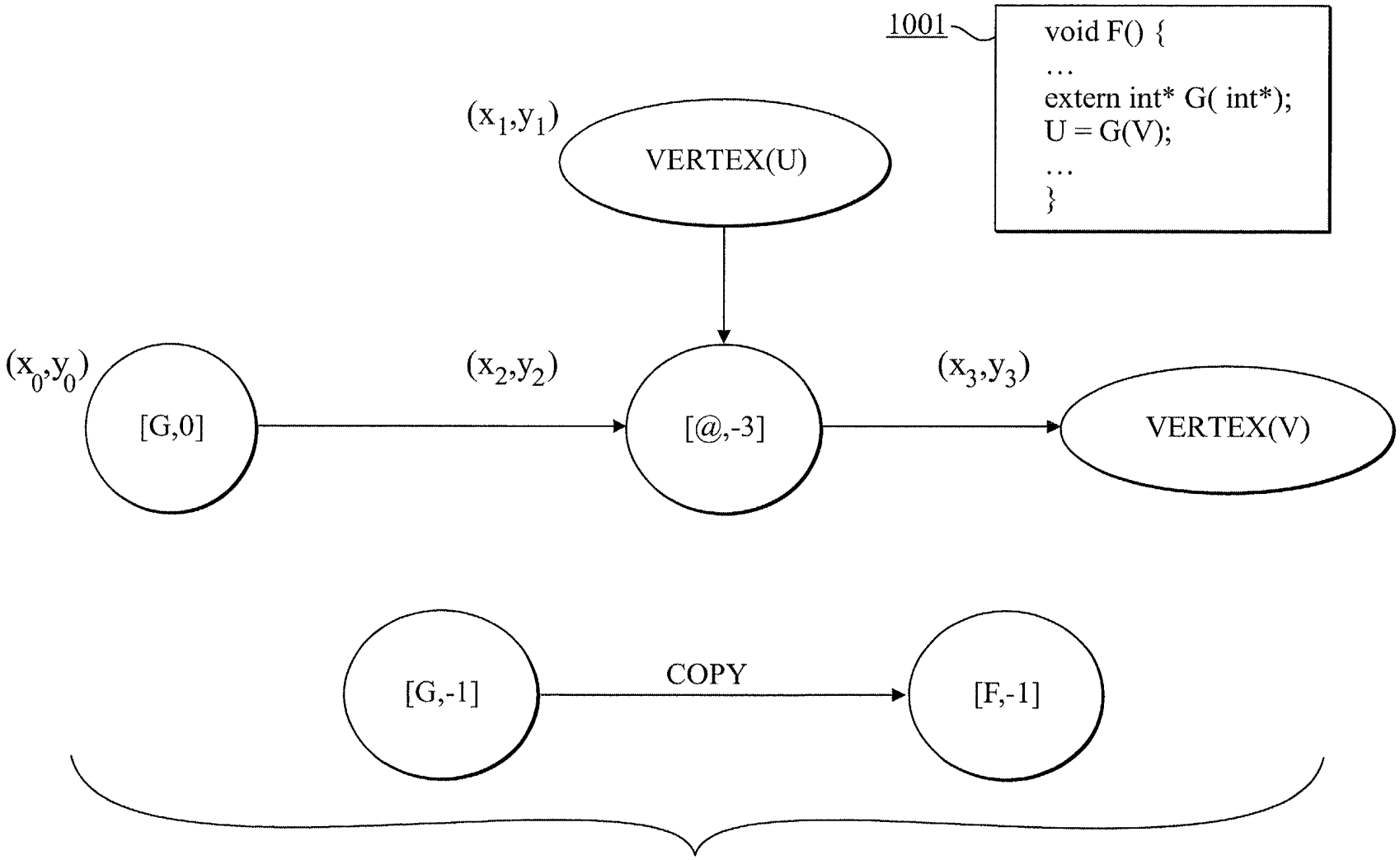


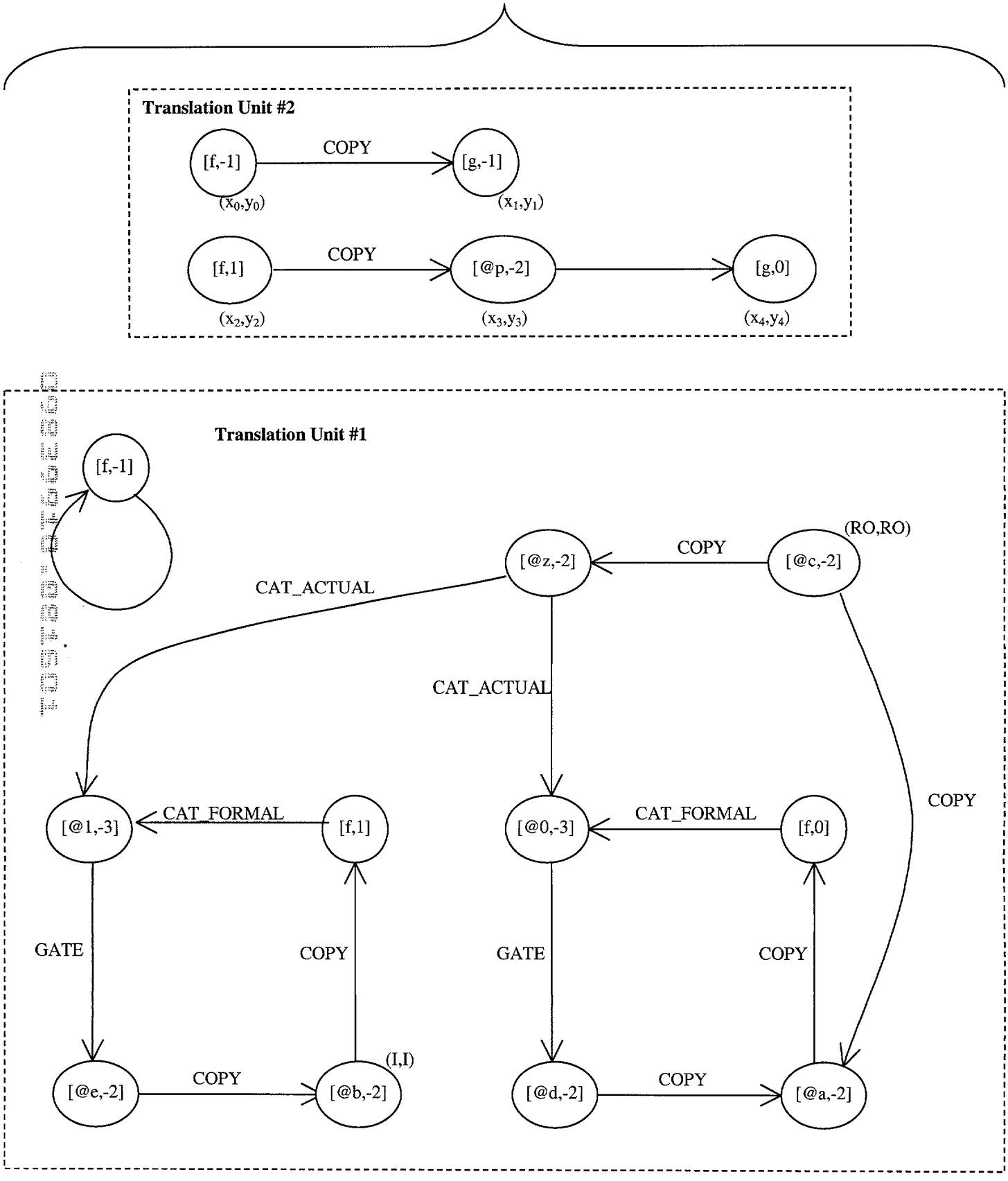
Fig. 10

```
// Translation unit #1

int* f( int* a, int* b, int n ) {
   int *c = a;
   if( n>0 ) {
      int* d = a+1;
      int* e = b+1;
      int* z = f( d, e, n-1 );
      c = z-1;
      *c = *b;
   }
   return c;
}
```

```
// Translation unit #2
extern int* f(int* a, int* b, int n );

void g( int* p ) {
   int y[10];
   f( &y[0], p, 10 );
}
```

FIG. 11

FIG. 12


Translation Unit #2
[f,-1]
COPY
[g,-1]
(x0,y0)
(x1,y1)
[f,1]
COPY
[@p,-2]
[g,0]
(x2,y2)
(x3,y3)
(x4,y4)
Translation Unit #1
[f,-1]
[@z,-2]
COPY
[@c,-2]
(RO,RO)
CAT_ACTUAL
CAT_ACTUAL
COPY
[@1,-3]
CAT_FORMAL
[f,1]
[@0,-3]
CAT_FORMAL
[f,0]
GATE
COPY
GATE
COPY
[@e,-2]
COPY
[@b,-2]
(I,I)
[@d,-2]
COPY
[@a,-2]

Translation Unit #1	
Source line	Action
(entry into f)	Add [@a,-2] →[f,0] Add [@b,-2] →[f,1]
int *c = a;	Add [@c,-2] → [@a,-2]
n>0	None
int *d = a+1;	Add [@d,-2] → [@a,-2]
int *e = b+1;	Add [@e,-2] → [@b,-2]
int* z = f(d,e,n-1)	Add [@z,-2] → [@0,-3] → [@d,-2] Add [f,0] → [@0,3] Add [@z,-2] → [@1,-3] → [@b,-2] Add [f,1] → [@1,3] Add [f,-1] → [f,-1]
c = z-1;	Add [@c,-2] → [@z,-2]
*c = *b;	Lower VAL[[@c,-2]] to O Lower VAL[[@b,-2]] to I
return c;	Lower VAL[[@c,-2]] to R

Translation Unit #2	
Action	**Action**
int *p = &x[0];	None
for(int i=0; i<10; i++)	None (no pointer assignments)
*p = i;	Lower VAL[[@p,-2]] to O
p=p+1	None (edge omitted by self-loop rule)
c = z-1;	Add [@c,-2] → [@z,-2]

FIG. 13